# An Unbiased Global Coin Flipping Protocol on Synchronous Distributed Systems

AUTHOR(S):

Yoda, Kunikazu; Okabe, Yasuo; Kanazawa, Masanori

CITATION:

Yoda, Kunikazu ...[et al]. An Unbiased Global Coin Flipping Protocol on Synchronous Distributed Systems. 数理解析研究所講究録 1996, 950: 63-70

ISSUE DATE:

1996-05

URL:

http://hdl.handle.net/2433/60340

数理解析研究所講究録
950巻 1996年 63-70



# An Unbiased Global Coin Flipping Protocol on Synchronous Distributed Systems

依田 邦和 (Kunikazu Yoda)†
岡部 寿男 (Yasuo Okabe)‡
金澤 正憲 (Masanori Kanazawa)‡
† Division of Applied Systems Science,
Faculty of Engineering, Kyoto University
‡ Data Processing Center, Kyoto University

**Abstract**

We present a distributed protocol for achieving totally unbiased global coin flipping in the presence of an adversary. We consider a synchronous system of $n$ processors at most $t$ of which may be corrupted and manipulated by a malicious adversary. We assume a complete network where every two processors are connected via a private channel.

Our protocol is deterministic and assumes a very powerful adversary. Although it cannot eavesdrop, it is computationally unbounded, capable of rushing and dynamic. This is the same model that is adopted in Yao's global coin flipping protocol [Yao84], which we use as the base of our protocol. Our protocol tolerates almost $n/3$ processor failures and terminates in $t+4$ rounds. The resilience of our protocol is greatly improved from that of Yao's protocol at the expense of running time, which is only added just two rounds.

## 1 Introduction

The distributed global coin flipping problem is to obtain a common random coin (which is a value in $\{0,1\}$) visible to all correct processors in the system despite malicious behavior of an adversary.

There are some cost measures of this protocol. The first one is time. In the synchronous system this is the number of rounds $r$ it takes during the execution of the protocol. The second parameter is resiliency. This is an upper bound on the number $t$ of faulty processors that can be tolerated compared to the number $n$ of all processors. The third parameter is message complexity. In general, this parameter is only concerned whether the numbers of bits grows polynomially or exponentially in $n$ and $t$. The power of the adversary (i.e. what resources it can access in the system) is also important in this protocol.

The coin flipping problem has been studied so that it can be used to implement or used inside the randomized Byzantine agreement protocol. There are actually two types of coin flipping, global and persuasive. The persuasive coin only requires that the sufficiently large part of the processors decide the same coin in nonzero probability, while the global coin requires that all processors decide the same coin with certainty.

The first protocol for the global coin flipping was proposed by Andrei Broder and Danny Dolev in 1984 [BD84], although this protocol required an encryption scheme and several conditions. A.C. Yao showed a deterministic global coin flipping protocol [Yao84] in 1984. He adopted a model in which the adversary cannot eavesdrop, but is computationally unbounded, capable of rushing, and dynamic. It requires $t \leq (n-1)/4$ and $t+2$ rounds. In 1990, S.Micali and T.Rabin showed a randomized protocol for achieving unbiased global coin flipping which requires $t \leq (n-1)/3$ and constant expected time. Their method is based on the VSS (Verifiable Secret Sharing).

Our protocol is deterministic and achieves unbiased global coin flipping with high resiliency. It is based on the Yao's protocol. It requires $t \leq (n-1)/3$ and $t+4$ rounds.

The remainder of this paper is organized as follows. In Section 2, we summarize the main aspects of our model, the adversary and the global coin flipping problem. In Section 3, we describe the history of the protocol. In Section 4, we show Yao's protocol which is the base of our protocol. In Section 5, our global coin flipping protocol is presented. In Section 6, the conclusion is provided.

## 2 Model and definitions

### 2.1 Basics of a synchronous distributed system

A *distributed system* is a set of processors that communicate by sending messages each other. All the processors are equal and no special leader exists. Some of the processors are faulty and behave maliciously.

We assume the following features of the distributed system.

- The communication network is complete.
- The communication link is bidirectional, reliable and private, i.e. each pair of the processors can send messages each other without an error nor being eavesdropped.
- Each processor knows the identity of the processor on the other end of each link.
- The processors don't share memory but each processor has enough local memory.
- *Correct* processors always follow the same prescribed protocol while *faulty* processors may not.

We assume a *synchronous* distributed system in which the following communication model is adopted.

- All the processors have completely synchronized clocks such that all the communications are done during time intervals (=*rounds*) defined by pulses of the clock. Any message sent between correct processors during one round will be received within that round. During one round, each processor executes the following three actions according to the protocol.
  - Send messages
    (A processor can send up to $m$ bits to each processor.)
  - Receive messages
    (Processors know who sent each of the messages.)
  - Do some internal computation
    (in general, to produce the next round's messages to be sent.)

## 2.2 The adversary

To explain types of faulty behavior, we suppose that the faulty processors are selected, corrupted and manipulated by an adversary.

In our protocol, the following abilities of the adversary are assumed.

- The adversary can corrupt and manipulate up to $t$ processors. Once corrupted, the processors are deprived of all input and output control by the adversary.
- Every message which is sent to or sent by a faulty processor is readable by the adversary.
- *Dynamic*: The adversary can corrupt processors at arbitrary point during execution.
- *Rushing*: The adversary can instruct faulty processors to "wait" until they get all $d$-th round messages from correct processors and then based on this examination, instruct the faulty processors on their own $d$-th round messages.
- *Computationally unbounded*: The adversary has unlimited computational power and can even read an encrypted message.

## 2.3 The unbiased global coin flipping problem

Roughly speaking, the global coin flipping problem is to combine the individual random sources of the processors in the system to obtain a random source visible to all the processors. The coin is totally unbiased when the probabilities of obtaining 0 and 1 are equal. The unbiased global coin flipping (UGCF) problem is stated as follows.

**Definition 1 (UGCF problem)** *Given are $n$ processors, at most $t$ of which may be faulty. Each processor $i$ secretly and randomly flips initial coins $\in \{0,1\}$ independent of the others. It is required to construct a protocol which satisfies the following conditions.*

- **Termination.** *Every correct processor $i$ eventually and certainly decides on a decision coin $d_i \in \{0,1\}$.*
- **Agreement.** *All correct processors decide on the same decision coin.*
- **Non-bias.** *The decision coin is absolutely unbiased, i.e.*

$$Pr(d_i = 0) = Pr(d_i = 1) = 1/2$$

*for all the efforts of an adversary.*

## 2.4 The single source Byzantine agreement

The Byzantine agreement (BA) protocol is generally used as a subroutine in the global coin flipping protocol to guarantee that the coin is agreed on among correct processors. We use another version of BA, *single source BA* as a subroutine of our UGCF protocol. Formally, the single source BA problem is defined as follows.

**Definition 2 (Single source BA problem)** *Given are $n$ processors, at most $t$ of which may be faulty. A source processor has an initial value $v \in \{0,1\}$. It is required to construct a protocol which satisfies the following conditions.*

- **Termination.** *Every correct processor $i$ eventually and certainly decides on a decision value $d_i \in \{0,1\}$.*
- **Agreement.** *All correct processors decide on the same value.*
- **Validity.** *If the source is correct , then $d_i = v$ for all correct processors $i$.*

The single source BA problem and the BA problem are practically identical and all the results mentioned in this paper apply to both problems with only minor modifications.

# 3 History of the problem

## 3.1 Cost measures

The global coin flipping protocol generally uses BA as a subroutine. In this approach the lower bounds of the global coin flipping depends on the lower bounds of BA.

Thus, this approach requires

- *resiliency*: $t \leq (n-1)/3$ [PSL80]
- *running time*: $r = t + 1$ rounds [DLM82]

The lower bound of running time is for deterministic protocols. For randomized protocols, there is a protocol which terminates in constant expected rounds such as [FM88].

## 3.2 Past results

There are a few protocols that achieves totally unbiased global coin flipping and assumes no eavesdropping, computationally unbounded, capable of rushing, and dynamic adversary on synchronous systems.

A deterministic protocol which assumes such a model was presented by A.C.Yao [Yao84] in 1984. His protocol requires

- $t \leq (n-1)/4$
- $r = t + 2$ rounds

An asynchronous version of Yao's protocol was presented by M.Ben-Or [Ben85].

A randomized protocol which assumes such a model was presented by S.Micali and T.Rabin [MR90] in 1990. Their protocol requires

- $t \leq (n-1)/3$
- constant expected rounds

Our protocol assumes the most simple case – deterministic protocol on synchronous systems, which is the same as Yao's protocol.

# 4 Yao's result

In this section, we describe the Yao's global coin protocol. His protocol assumes a powerful adversary. Although it cannot eavesdrop, it is computationally unbounded, capable of rushing, and dynamic. His protocol requires $t \leq (n-1)/4$ and $t+2$ rounds, and the message complexity is exponential.

**Yao's UGCF protocol:**

1. Partition the $n(\geq 4t+1)$ processors into all $\binom{n}{3t+1}$ groups of $3t+1$ processors. Each processor belongs to $\binom{n-1}{3t}$ groups simultaneously.
2. Each processor flips a initial coin in $\{0,1\}$ independently and randomly for each of the $\binom{n-1}{3t}$ groups of which it is a member.
3. Within each of the $\binom{n}{3t+1}$ groups, $3t+1$ separate single source BA protocols are executed concurrently. The sources' initial values for each of the single source BA protocols are {0,1}s fliped in Step 2.
4. Each processor in the group computes the XOR of the $3t+1$ values agreed on in Step 3 and stores that value as the group coin.
5. Each processor in the group broadcasts the value of the group coin.
6. Each processor recieves $3t+1$ group coins for each of the $\binom{n}{3t+1}$ groups and stores each majority value of these coins as each group coin.
7. The global coin is the XOR of $\binom{n}{3t+1}$ group coins.

(Note: Step 4 and 5 are done for each group of which the processor is a member.)

**Lemma 1** *Yao's protocol achieves unbiased global coin flipping for* $n \geq 4t+1$, $r = t+2$ *and exponential message size.*

*Proof:* It is obvious that the termination condition holds.

**Agreement.** Faulty processors are less than one-thirds of the processors within each group, so single source BA can be executed within each group in Step 3. Because of the properties of single source BA, all the correct processors within the group see the same set of $3t+1$ outcomes, so they agree on the same group coin in Step 4. Since at least $2t+1$ of the $3t+1$ members of a group are correct, when a processor which is not the member of the group receives values from processors in the group in Step 6, the majority of these value are the actual value agreed on by the correct members of the group, despite the messages from the faulty processors. So all the correct processors see the same set of group coins in Step 6. Therefore the XOR of these group coins, which is the decision coin, is common to all correct processors.

**Non-bias.** We choose every combination of $3t+1$ processors out of $n(\geq 4t+1)$ processors. The number of the faulty processors is at most $t$, so there is at least one group which does not have any faulty processor, regardless of when processors are corrupted. Such a group is called a *pure* group. The point at which all $\binom{n}{3t+1}$ group coins are determined (at the end of Step 4) is called a *turning point.* It is obvious that the coin of the pure group is uniformly random. The coin of the pure group is independent of all other group coins because the adversary cannot eavesdrop on the messages between the members of the pure group, and thus the behavior of the adversary before the turning point is determined with no information about the coin of the pure group. After the turning point, the adversary cannot change any group coin, nor can it prevent group coins from being revealed to the other groups. The XOR of group coins, at least one of which is uniformly random and independent of the others, is itself uniformly random. Thus the non-bias condition holds.

## 5 Our protocol

We adopt the same model and the same type of adversary that is used in Yao's protocol. Our protocol improves the resiliency compared with Yao's protocol, while it needs just two more extra rounds than the round required in Yao's protocol. Our protocol requires $t \leq (n-1)/3$, and $t+4$ rounds and exponential message size.

**Protocol:**

1. Partition the $n(\geq 3t+1)$ processors into all $\binom{n}{2t+1}$ groups of $2t+1$ processors. Each processor belongs to $\binom{n-1}{2t}$ groups simultaneously.

2. Each processor flips an initial coin in {0,1} independently and randomly for each of the $\binom{n-1}{2t}$ groups of which it is a member.

3. Each processor executes the following two-round algorithm concurrently for each of the $\binom{n-1}{2t}$ groups of which it is a member.

   Round 1: Send the initial coin for this group to every processor in this group. Receive $2t+1$ coins from the members of the group. Calculate the XOR of these coins.

   Round 2: Send the XOR calculated in round 1 to every processor in the group. Receive $2t+1$ values from the members of the group. If all these values are the same, store D (decided) and the value {0,1}, otherwise, store U (undecided).

4. For each of the $\binom{n}{2t+1}$ groups, $2t+1$ separate single source BA protocols are executed concurrently among all processors in the system. The sources' initial values are {U,D}s determined at the end of Step 3.

5. At this point, every correct processor has $2t+1$ values of {U,D} agreed on among all the correct processors in the system for each of the $\binom{n}{2t+1}$ groups. For each group, if all these $2t+1$ values are Ds, decide D as the group value, otherwise decide U as the group value.

6. If a processor belongs to the group whose group value was D in Step 5, it broadcasts the coin {0,1} stored in Step 3 as the group coin. If a processor receives coins from the processor whose group value was U in Step 5, it ignores them.

7. Every processor receives $2t+1$ coins for each group whose value was D in Step 5, and stores each majority of these coins as each of group coin.

8. Each processor calculates the XOR of these group coins, which is the decision coin.

**Lemma 2** *The protocol achieves unbiased global coin flipping for* $n \geq 3t+1$, $r = t+4$ *and exponential message size.*

*Proof:* It is obvious that the termination condition holds.

**Agreement.** We choose every combination of $2t+1$ processors out of $n(\geq 3t+1)$ processors. The number of faulty processors is at most $t$, so there is at least one group

which does not have any faulty processor, regardless of when processors are corrupted. Such a group is called a pure group.

In Step 3, the processors in the pure group always decide D and store the same coin {0,1}. The correct processors in the group which contains faulty processors may decide U or D, but the coin {0,1} of the correct processors which decide D are the same within the group.

Because faulty processors are less than one-thirds of the processors in the system, single source BA can be executed in the system in Step 4.

At the end of 4, every correct processor has each group values of {U,D} of each processor (including faulty processors) (totally, $\binom{n}{2t+1} \times (2t+1)$ [bit]). Because of the properties of single source BA, all these values are the same among correct processors and so are the values of {U,D} determined in Step 5. All correct processors always determine D as the pure group value.

The correct processors of the group whose value is D in Step 5 always store D in Step 3, because if one of them stores U in Step 3, U is determined in Step 5. Their coins {0,1} are common to the correct processors within the group

Since a group consists of $2t+1$ processors and faulty processors are at most $t$, the majority of the group are correct. To decide a group coin, a correct processor calculates the majority of the coins received from that group, so the same coin that the correct processors of the group hold is determined despite the coins from the faulty processors.

All correct processors can see the same set of groups which is determined D in Step 5 and their coins (group coins). Such a group always exists, for example the pure group.

Therefore the XOR of these group coins, which is the decision coin, surely exists and it is common to all correct processors.

**Non-bias.** Look at the point at which single source BA protocols have just finished (at the end of 5). That point is called a turning point. The group coins which will be used for the decision coin are determined at this point.

It is obvious that the coin of the pure group is uniformly random. The coin of the pure group is independent of all other group coins because the adversary cannot eavesdrop on the messages between the members of the pure group and single source BA protocols are executed with values {U,D} which have no information about the initial coins of the pure group, thus the behavior of the adversary before the turning point is determined with no information about the coin of the pure group.

After the turning point the adversary has no way to change these values of {U,D} and it cannot prevent the group coins whose group is determined D at the turning point from being revealed to the other groups.

The decision coin is the XOR of the group coins whose group is determined D at the turning point. At least one of these coins is uniformly random and independent of the others, so the decision coin itself is uniformly random. Thus the non-bias condition holds.

**running time.** Protocols of different groups can be executed concurrently:

- It takes two rounds in Step 3.
- Single source BA protocol takes $t+1$ rounds.
- It takes one round to reveal group coins to the other groups.

So, totally it takes $t+4$ rounds.

## 6 Conclusion

We showed an unbiased global coin flipping protocol which tolerates almost one-thirds of the processor failures and requires $t + 4$ rounds.

Although our protocol is designed for synchronous systems, we hope that an asynchronous version of this protocol is also possible.

## References

[BD84] A. Z. Broder and D. Dolev. "Flipping coins in many pockets". pages 157–170, 1984.

[Ben85] M. Ben-Or. "Fast asynchronous Byzantine agreement". pages 149–151, 1985.

[CD89] B. Chor and C. Dwork. "Randomization in Byzantine Agreement". In S. Micali, editor, *Advances in Computer Reaserch, Randomness and Computation*, volume 5, pages 443–497. JAI Press, 1989.

[DLM82] R. DeMillo, N. A. Lynch, and M. Merritt. "Cryptographic Protocols. In *Proceedings* $14^{th}$ *Annual ACM Symposium on Theory of Computing*, pages 383–400, 1982.

[DSS86] C. Dwork, D. Shmoys, and L. Stockmeyer. "Flipping Persuasively in Constant Expected Time". *Proceedings of the* $27^{th}$ *Annual Symposium on Foundations of Computer Science"*, pages 222–232, 1986.

[FM88] P. Feldmon and S .Micali. "Optimal Algorithms for Byzantine Agreement". pages 148–161, 1988.

[LSP82] L. Lamport, R. Shostak, and M. Pease. "The Byzantine Generals Problem". *ACM Transactions on Programming Languages and Systems*, pages 382–401, 1982.

[MR90] S. Micali and T. Rabin. "Collective Coin Tossing Without Assumptions nor Broadcasting". *Advances in Cryptology - CRYPTO*, pages 253–266, 1990.

[PSL80] M. Pease, R. Shostak, and L. Lamport. "Reaching Agreement in the Presence of Faults". *Journal of the ACM*, pages 228–234, 1980.

[Yao84] A. C. Yao. public lecture. 1984.